Q1) Differentiate Between Informed & Uninformed search.

=>

Informed search	Uninformed search
1) It is also known as Heuristic search.	1) It is also known as Blind search.
2) It uses knowledge for the searching process	2) It doesn't use knowledge for the searching process.
3) It finds a solution more quickly	3) It finds solution slow as compared to an informed search.
4) It may or may not be complete	4) It always complete.
5) Cost is low	5) Cost is high
6) It consumes less time because of quicle searching	6) It consumes moderate time because of slow searching
7) There is a given direction about the solution.	7) No suggestion is given regarding the solution in it

Q2) Explain Iterative Deepening DFS algorithm with an example.

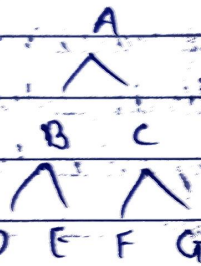
=> Search algorithm that combines space efficiency of DFS & completeness & optimality of BFS.

- It finds out best dept limit by gradually increasing limit until goal is found.

Instead of searching deeply first (DFS) or level

By level (BFS) ; IDDFS repeatedly runs depth limited search.
It start with depth limit 0 & increase gradually.
Time complexity : $O(b^d)$
space complexity: $O(d)$
completeness : yes
optimality : yes

eg

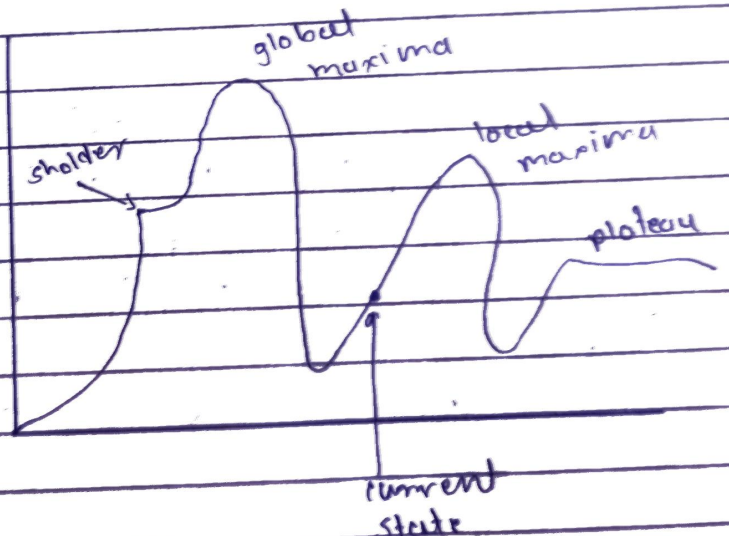


suppose we want to find node G starting from node A using IDFS.

1) Start with a depth limit of 0. Performs DFS starting from node A.
2) Increase the depth limit to 1 & perform DFS again, but only expand nodes upto depth limit 1.
3) Increase the depth limit to 2 & repeat the process until the goal node G is found

(Q 3) Explain Hill Climbing algorithm, its types & drawbacks in detail.

=) Algorithm generally moves up in direction of informed search algorithm increasing value.
continuously moves towards the direction of increasing value until it reaches peak.
used in optimisation problem
goal is to find best possible solution.
It does not maintain search tree, similar to greedy local search.
It starts with initial state, evaluates neighbour state & choose one with best value.
Terminates when no better neighbouring state found.



Types i) Simple Hill Climbing:
consider only immediate best option with backtracking.

ii) Steepest - ~~Hill~~ Ascent Hill Climbing:

11) Stochastic Hill climbing :- Randomly selects better move instead of always choosing best.

Dis-adv

1) ⚠ due to sholder can't see the way out.
2) local maxima problem :- algorithm may stop at peak that is not global maxima.
3) ridges traveller : may struggle with sharp slopes or deep valley's.
4) plateau problem : all neighbouring states have same value.
5) backtracking is not supported.

(Q4) Explain A* with suitable example.

=) A* is a popular informed search algorithm used for finding the shortest path in a graph.
- It combines the strengths of Dijkstra's Algorithm & Greedy Best first search.
- It uses the following cost functions.

$$f(n) = g(n) + h(n)$$

where,

f(n) - Total estimated cost of path though node.
g(n) → Actual cost from start node to node n.
h(n) → Heuristic estimate of cost from node n to goal.

ex.

Node	g(n)	h(n)	f(n)
S	0	6	6
A	2	4	6
B	4	2	6
G	6	0	6

steps

1) start at S

- openlist : [S] closed list [ ]
- Expand S : Add A & B to openlist

2) Evaluate A

$g(A) = 2$, $h(A) = 4$, $f(A) = 6$

3) Evaluate B

$g(B) = 4$, $h(B) = 2$, $f(B) = 6$

4) Select A & expand it.

Add G

$g(G) = 6$, $h(G) = 0$, $f(G) = 6$.

5) Goal G is reached with total cost 6.